Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/5510--07-9022

Integration of Two SPAWAR PEOC4I NetCentric Technologies: Tactical Environmental Database Services (TEDServices) With the Extensible Tactical C4I Framework (XTCF)

Timothy H. Bowers

Navy Center for Applied Research in Artificial Intelligence
Information Technology Division

January 12, 2007

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 12-01-2007	**2. REPORT TYPE** Memorandum Report	**3. DATES COVERED** *(From - To)* May 2005 - Oct 2006

4. TITLE AND SUBTITLE Integration of Two SPAWAR PEOC4I NetCentric Technologies: Tactical Environmental Database Services (TEDServices) with the Extensible Tactical C4I Framework (XTCF)	**5a. CONTRACT NUMBER** N0001406WX20076
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 0602235N
6. AUTHOR(S) Timothy H. Bowers	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER 55-7188
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory, Code 5513 4555 Overlook Avenue, SW Washington, DC 20375-5320	**8. PERFORMING ORGANIZATION REPORT NUMBER** NRL/MR/7210--07-9022
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research 875 North Randolph Street Arlington, VA 22203-1995	**10. SPONSOR / MONITOR'S ACRONYM(S)**
	11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This paper outlines work that was completed to assist the warfighter during the critical mission planning process. This was accomplished by delivering current weather data from Tactical Environmental Data Services (TEDServices), an API used to request meteorological, oceanographic, and environmental information, through the Extensible Tactical C4I Framework (XTCF), which is a prototype extensible data management framework implemented in Java. It includes discussion of relevant technologies, such as XML and JMS.

15. SUBJECT TERMS

NetCentric	TEDServices	JMS
XTCF	XML	

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Timothy H. Bowers
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UL	27	**19b. TELEPHONE NUMBER** *(include area code)* (202) 404-3924

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

Section **Page**

1 INTRODUCTION 1-1

2 RELEVANT TECHNOLOGIES 2-2

2.1 EXTENSIBLE MARKUP LANGUAGE (XML) 2-2

2.2 JAVA MESSAGE SERVICE (JMS) 2-3

3 ARCHITECTURE 3-4

4 SYSTEM REQUIREMENTS 4-5

4.1 XTCF SYSTEM REQUIREMENTS 4-5

4.2 TEDSERVICES SYSTEM REQUIREMENTS 4-6

5 INSTALLATION 5-6

5.1 INSTALL TEDSERVICES [5] 5-6

5.1.1 Install SSL Certificate [5] 5-7

5.2 INSTALL XTCF 5-7

5.2.1 JBoss Setup [7] 5-8

5.2.2 Setting Delayed Expansion [7] 5-9

5.2.3 Building XTCF [7] 5-9

5.3 INSTALL PLUGINS 5-9

6 WORK FLOW 6-10

7 DATA 7-15

8 TROUBLESHOOTING TIPS 8-17

9 REFERENCES 9-18

APPENDIX A. ACRONYM LIST 9-19

APPENDIX B. EVISMESSAGE.JAVA 9-20

TABLE OF FIGURES

Figure **Page**

Figure 2-1. Java Message Service (JMS) [6] 2-3
Figure 3-1. Architecture Diagram 3-5
Figure 6-1. JBoss Started 6-11
Figure 6-2. Reply Plug-In Listens for Request Messages 6-11
Figure 6-3. EVIS Request GUI 6-12
Figure 6-4. Request Plug-In Sends Request Message 6-13
Figure 6-5. EVIS Receives and Returns TEDs Data 6-14
Figure 6-6. TEDs Data Received and Displayed in GUI 6-15
Figure 7-1. Grid Data Display 7-16

Integration of two SPAWAR PEOC4I NetCentric Technologies: Tactical Environmental Database Services (TEDServices) With the Extensible Tactical C4I Framework (XTCF)

1 INTRODUCTION

We are currently living in the information age. Information is readily available on the web and can be accessed via web pages and web services. People log-on to the internet everyday searching for information. With all the search engines and websites available it can take hours to complete this search. Not everyone has the luxury of spending hours on the internet searching for data. In order to make time critical decisions, they need accurate information and they need it in a timely manner. For example, the warfighter needs to have access to various types of information in order to complete time critical tasks. This paper outlines work that was completed, which developed plug-ins to assist the warfighter during the critical mission planning process by delivering current weather data.

The first step in accomplishing this is to obtain useful and accurate weather information. Obtaining information from another computer can be performed by the use of a gateway, which controls access to the information by ensuring certain protocols and procedures are followed. Tactical Environmental Data Services (TEDServices) has created an Application Program Interface (API) that allows users to connect to a TEDServices gateway to request meteorological, oceanographic, and environmental information for various parameters. The plug-ins outlined in this paper use this API and connect to a TEDServices gateway to retrieve data.

After the data has been obtained, it needs to be delivered to the warfighter in an efficient and reliable manner. It is possible to communicate information by using a messaging technology. JAVA has developed an API for messaging called Java Messaging Services (JMS), which is discussed in more detail in section 2.2. The plug-ins, described in this paper, use the JMS API via the Extensible Tactical C4I Framework (XTCF). XTCF is a prototype extensible data management framework implemented in Java for the basic distribution of information objects to the warfighter using core, standards-based discovery, subscription, and publication building blocks of the Web Services model. The goal of the XTCF project is to provide the warfighter with a flexible, extensible framework that could support near real time delivery of information composed to meet specific mission requirements. The capabilities of this prototype were displayed in a limited technology experiment, which was conducted in June 2004. The results from this experiment are published in Filanowitcz [3].

Manuscript approved November 8, 2006.

2 RELEVANT TECHNOLOGIES

This section provides an overview of relevant technologies that may be applicable when developing XTCF and TEDServices plug-ins.

2.1 EXTENSIBLE MARKUP LANGUAGE (XML)

XML, the Extensible Markup Language, is a W3C-endorsed standard for document markup [4]. This standard allows for the markup of data with human-readable tags which are similar to html. It provides a standard format that can be customized for various domains and can be used to exchange data on and off the web. It has become ubiquitous and a variety of tools are readily available for generating, parsing, and processing XML data [8]. In general XML is made up of nested elements that begin with a start tag (<ElementName>) and end with an end tag (</ElementName>), which is shown below.

.

```
<address>
   <street>1234 Maple Street</street>
   <City>Atlanta</City>
   <State>Georgia</State>
   <Zipcode>30301</Zipcode>
</Address>
```

The flexibility of XML allows users to define the structure of their documents. This structure is defined in an XML Schema Document (XSD) file. The following is an example of an XSD file that was used with these plug-ins.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:evis.samples.xtcf"
  xmlns="urn:evis.samples.xtcf">
  <xs:complexType name="EVISMessage">
    <xs:sequence>
      <xs:element name="FileName" type="xs:string" default=""/>
      <xs:element name="FileContent" type="xs:hexBinary" default=""/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="EVISMessage" type="EVISMessage"/>
</xs:schema>
```

Messages in XTCF can be defined by an XSD. The XTCF SDK uses a utility called codegen. Codegen is a command line utility that will translate XML Schema based messages into code [8]. It is a java program that is executed by running a batch file that when executed creates java code from the XSD. More information on using codegen can be found in the Developers Guide for the XTCF SDK. The file that was created using codegen on the above XSD can be found in Appendix B.

2.2 Java Message Service (JMS)

The Java Message Service is a Java API implemented by enterprise messaging vendors to provide Java applications with a common and elegant programming model that is portable across messaging systems [2]. This API defines a common set of programming strategies that will support all JMS compliant messaging systems. It is a powerful tool that allows applications to exchange critical information in a reliable and asynchronous manner. Therefore, Messages can be delivered to systems not currently running and processed when convenient [8]

Figure 2-1 shows the relationship of the classes and interfaces in the Java Message Service (JMS) API. Developers use these classes and interfaces to create a JMS application [6]. The finished application will need to create a connection, which allows a producer and consumer to send messages back and forth using the JMS API.

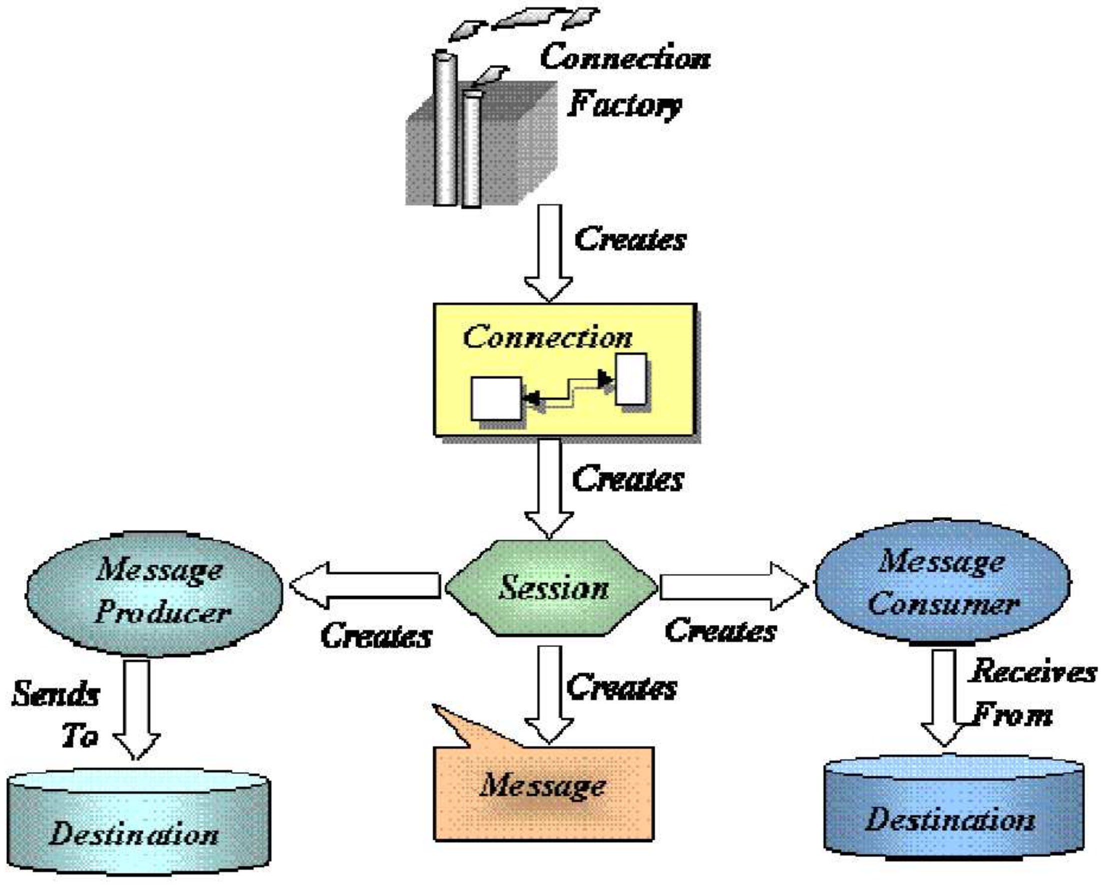


Figure 2-1. Java Message Service (JMS) [6]

There are two common programming models supported by the JMS API: publish-and-subscribe and point-to-point [2]. The plug-ins that were developed for this research use the publish-and-subscribe programming model. This model uses a virtual channel called a topic to exchange messages. The publisher sends a message on a specific topic. Then any plug-in that subscribes to this topic can receive the message. The point-to-point programming model uses queues. A message posted to a queue is only intended for a single consumer [8].

In XTCF, a messaging system is used to provide the necessary asynchronous exchange of messages between plug-in components [8]. A messaging system is referred to as a broker. A free version of JBoss was used as the broker for the development of these plug-ins, but earlier versions of these plug-ins were tested using JBoss, SonicMQ, and BEA WebLogic.

3 ARCHITECTURE

The plug-ins have ten main files, two for the messages (MetOcRequestMessage.java and MetOcMessage.java), five for the MetOc Data Consumer (MetOcRequestGUI.java, MetOcRequestSender, MetOcReplyReceiver, MetOcReplyListener and MetOcReplyGUI), and three for the MetOc Data Provider (MetOcRequestReceiver.java, MetOcRequestListener.java and MetOcReplySender). The relationship and interactions of these files are depicted in figure 3-1. The MetOc Data Consumer sends a request for data to the MetOc Data Provider. When the MetOc Data Provider receives this request, it sends a request for the same data to TEDServices and then waits for a reply. Once the MetOc Data Provider has received the data from TEDServices, it then sends this data to the MetOc Data Consumer.

The request for data from MetOc Data Consumer is sent in a MetOcRequestMessage object on a METOCREQUEST topic. When the data is returned it is in a MetOcMessage object, which was sent on the METOCREPLY topic. MetOcMessage.java was automatically generated using XTCF codegen Version 1.2. It is used to return the data that is received from TEDServices. The same file was edited and renamed (MetOcRequestMessage) to send requests for MetOc data. The MetOcMessage and MetOcRequestMessage class objects represent Java runtime instantiations of XTCF messages.

When you build the java files using ANT, two batch files are placed in the C:\XTCF\bin directory. One is called runMetocDataConsumer and runMetocDataProvider, which run the plug-ins. Before using them you need to ensure that the JMS (JBoss) provider is set up correctly and running.

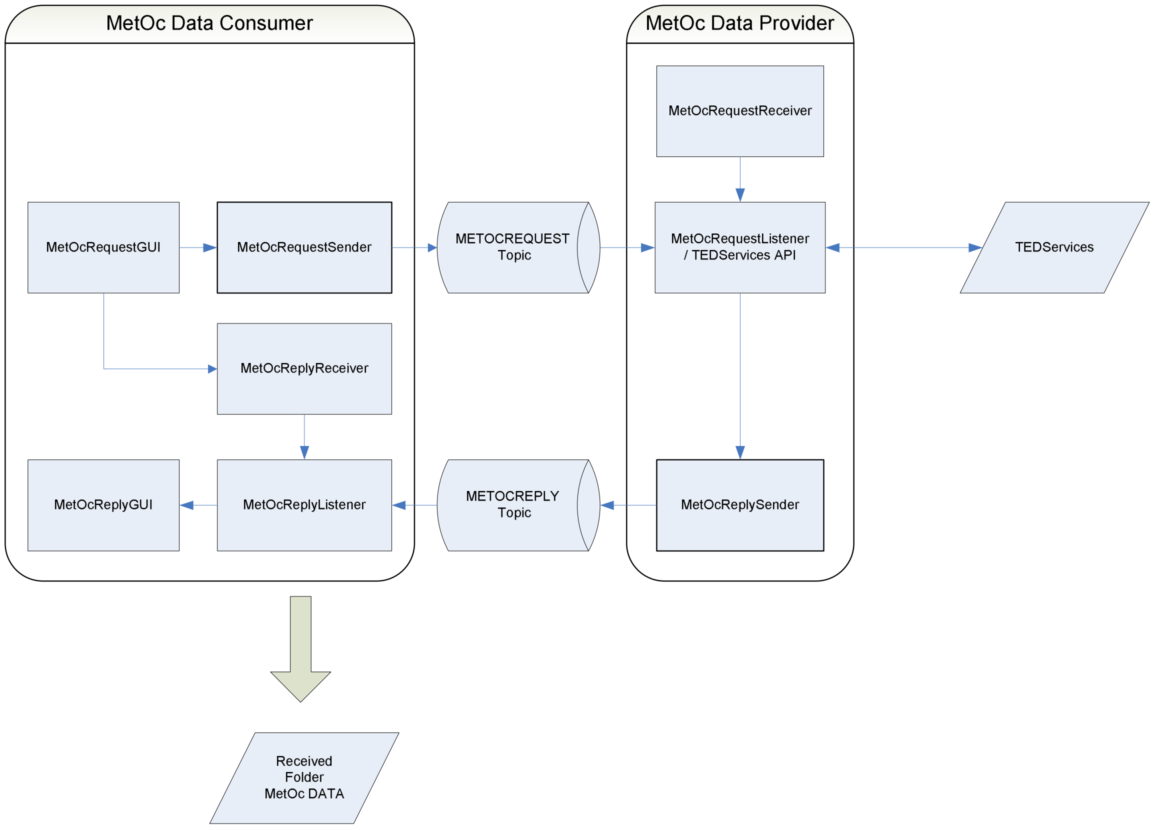


Figure 3-1. Architecture Diagram

4 SYSTEM REQUIREMENTS

This section provides an overview of the system requirements for the plug-ins.

4.1 XTCF System Requirements

Obtain a copy of the XTCF SDK from the following:

Space and Naval Warfare System Center, San Diego

Joint Tactical Information Systems Branch (Code 2734)

53560 Hull Street

San Diego, CA 92152-5001

Also, the XTCF development environment requires the following:

- Java Message Service (JMS; the XTCF Team has tested with SonicMQ version 5 and has done limited testing with JBoss version 4.0 and BEA WebLogic 8.1)
- Java SDK v1.4 – with JAVA_HOME environmental variable set to the location of JAVA SDK
- Jakarta ANT v1.5.1 – with ANT_HOME environmental variable set to the location of ANT
- MySQL v4.0.12 (optional for use with plug-in specific data)
- JMS v1.1 – with JMS_HOME environmental variable set to the location of JMS

4.2 TEDServices System Requirements

The machine on which the client application is developed must have installed, at a minimum, the TEDServicesAPI.jar and JDK 1.4.1. The TEDServicesAPI.jar file can be downloaded at https://teds.navy.mil. Sample code, documentation, and the TEDServicesAPI.jar file are included in the TEDServicesClientApplicationDeveloperToolKit, which can be downloaded from the site as well.

5 INSTALLATION

First obtain the distributions for TEDServices, XTCF, and EVIS Plug-ins. Then, follow the installation instructions below. If you have any problems installing the SDKs and plug-ins, review the troubleshooting tips section for assistance.

5.1 Install TEDServices [5]

1. Install the JDK 1.4.1, which can be obtained at: http://java.sun.com/j2se/1.4.1/download.html
2. Place the downloaded TEDServices API jar file in the directory where you would like it to permanently reside. (File initially resides in the directory: TEDServicesClientApplicationDeveloperToolKit\TEDServicesAPI) For the development of the plug-ins TEDServicesAPI.jar was placed in the C:\XTCF\lib\tedservices directory.
3. Alter your classpath to include a pointer to the TEDServices API jar file.
4. The Fleet MetOc (Meteorological and Oceanographic) Advanced Concepts Laboratory (FMACL) Developer TEDServices GateWay, 205.67.220.9, was used to test the plug-ins. The REMOTE_GATEWAY variable is set in the batch file that runs the plug-ins. To use this GateWay you need to import the supplied SSL certificate into your JRE certificate database (keystore). To do this follow the instructions outlined in section 5.1.1.

5. Compile and run the sample application provided at: TEDServicesClientApplicationDeveloperToolKit\SampleCode\SystemTestApi.java.

   A script file is provided in that same directory to run the sample code. The script file will require editing some paths relative to your machine. A script file has been provided for Windows (RunSystemTestApi.bat) and a script file has been provided for Linux (RunSystemTestApi).

This completes successful set up and testing of necessary components. You are now ready to integrate your own application with the TEDServices API.

5.1.1 Install SSL Certificate [5]

All information sent and received to/from TEDServices is sent via a form of http called https. Https is almost identical to http, but allows for secure communication using encryption and authentication. This encryption/authentication is performed by the secure socket layer (SSL). In order to communicate with TEDServices in a secure manner you need to install a SSL certificate.

In Windows use:
"keytool -import -alias tedservices -keystore <PathToJRE>\lib\security\cacerts -file <RemoteGatewayCert>\tedservices.cert -storepass changeit"

In the above command, <PathToJRE> should be replaced with the actual location of the JRE install and <RemoteGatewayCert> should be replaced with the actual location of the remote Gateway’s public certificate (tedservices.cert).

Example: "keytool -import -alias tedservices -keystore C:\jdk1.4.1\jre\lib\security\cacerts -file c:\tedservices.cert -storepass changeit"

Note: Keytool is a java command located in the <PathToJRE>\bin. If the System path is set, then the keytool command can be accessed without specifying the whole directory path. If the system path is not set, then the absolute path should be specified when calling Keytool.

Example: "c:\jdk1.4.1\jre\bin\keytool -import -alias tedservices -keystore c:\jdk1.4.1\jre\lib\security\cacerts -file tedservices.cert -storepass changeit"

5.2 INSTALL XTCF

There are many ways to install XTCF. The following is how XTCF was installed for these plug-ins. First, you need to copy the XTCF-Tools, XTCF-Docs, and XTCF folders to the root directory. Set XTCF_HOME environmental variable set to the location of XTCF. Then you will need to configure JBoss, set delayed expansion, and build XTCF and the plugins.

5.2.1 JBoss Setup [7]

Perform the following steps to configure the JBoss JMS provider.

Prerequisites:

1. Install JBoss on all clients and servers. Open a ***Windows Explorer*** window and navigate to the **XTCF-Tools** folder under **XTCF**. In the **Downloads** folder open the jboss-4.0.0DR2.zip file and extract it to C:\jboss-4.0.0DR2.
2. Create JBOSS_HOME environment variable. Set JBOSS to where it is installed on local machine or where JBoss JMS Client JARs are locally installed.
3. Increase the JVM memory by changing the JAVA_OPTS line in the JBoss run.bat file. Edit the run.bat in the JBoss bin folder. Modify an existing line by replacing **rem** with **set** in the set JAVA_OPTS=%JAVA_OPTS% -Xms256m -Xmx512m line.

Step 1-Add Users.

To create the appropriate users for JBoss, copy the jbossmq-state.xml file from the ${XTCF_HOME}\data\Config\JBoss\server\default\conf\ directory to the following location:

${JBOSS_HOME}\server\default\conf\jbossmq-state.xml.

Step 2-Create topics and queues.

To create the appropriate topics and queues for JBoss, copy the following files from the ${XTCF_HOME}\data\Config\JBoss\server\default\deploy\ directory to the ${JBOSS_HOME}\server\default\deploy\ directory:

xtcf-elint-rptmgr-service.xml

xtcf-elint-trkmgr-service.xml

xtcf-xema-service.xml

xtcf-xex-service.xml

xtcf-gale-service.xml

xtcf-midb-service.xml

xtcf-repository-service.xml

This causes JBoss to read and create the queues and topics automatically.

Note: Note these topics and queues are for the various plugins that are included in the XTCF SDK. This procedure will be repeated for xtcf-evis-service.xml when you install the plugins in section 5.3.

Step 3- In a ***command*** window run the setJBoss.bat or setJBoss.sh file in the XTCF/bin directory.

Note:	New JNDI properties do not need to be edited if running on localhost, otherwise edit the jndi.properties file.

Step 4-Start JBoss.

Execute the run.bat file in the ${JBOSS_HOME}\bin directory.

5.2.2 Setting Delayed Expansion [7]

Before running the XTCF .bat files, ensure that the Windows system is enabled to delay expansion of variables. This allows batch files to use environmental variables instead of hard coded path names for access to commands or files. Run **regedit** in a ***Command Prompt*** window. On the left-hand side of the window that appears, navigate to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Command Processor

In the right window pane, look for a DelayedExpansion entry. If it does not appear, add it by right clicking in the pane and creating a new DWORD entry. Edit the entry by double clicking on it and set the data value to 1 (true).

5.2.3 Building XTCF [7]

The easiest way to compile and build XTCF is to open a ***Command Prompt*** window and execute the XTCF-Setup.bat file from the XTCF-Tools folder. This will set up the build time environment. The build needs two environment variables: XTCF_TGT and JMS_HOME. The XTCF-Setup.bat file will set the JMS_HOME variable to the XTCF_Tools\JMS1.1 folder, and it will set XTCF_TGT (the destination folder of the build) to C:\XTCF. If a developer uses another version of the JMS JAR files or wishes to change the destination folder location.

Note:	Ant must be installed and configured on the build machine to build XTCF. Running the XTCF-Setup.bat file in compiling and building XTCF configures the system to use the jakarta-ant-1.5.1 provided in the XTCF-Tools folder.

After running XTCF-Setup.bat file, go to the location of the XTCF source tree: XTCF-SRC\xtcf. Type **ant**. Next go to the XTCF-SRC\xtcf-plugins directory. Type **ant**. Note the Ant output in the ***Command Prompt*** window. A successful build will create class files in the XTCF-SRC\xtcf\build and the XTCF-SRC\xtcf-plug-ins\build folders. Also, the build output will populate C:\XTCF.

5.3 INSTALL PLUGINS

Obtain the Evis files (which will be in one folder called Evis), build.xml file, and xtcf-evis-service.xml file from the author of this paper. Then perform the following steps.

Step 1-Create directories and copy files.

Create the following directory, C:\XTCF\XTCF-Plugins\xtcf-plugins\evis and copy the entire contents of the evis folder into this directory. Next rename the build file (build.xml), located at C:\XTCF\XTCF-Plugins\xtcf-plugins, old.build.xml.old. Then copy the build file build.xml from the evis disk to C:\XTCF\XTCF-Plugins\xtcf-plugins.

Step 2-Create topics and queues.

Create the appropriate topics for evis by copying the xtcf-evis-service.xml file to the ${JBOSS_HOME}\server\default\deploy\ directory. This will allow JBoss to read and create the evis topics automatically.

Note: The xtcf-evis-service.xml file was created by altering one of the xtcf-???-service.xml files

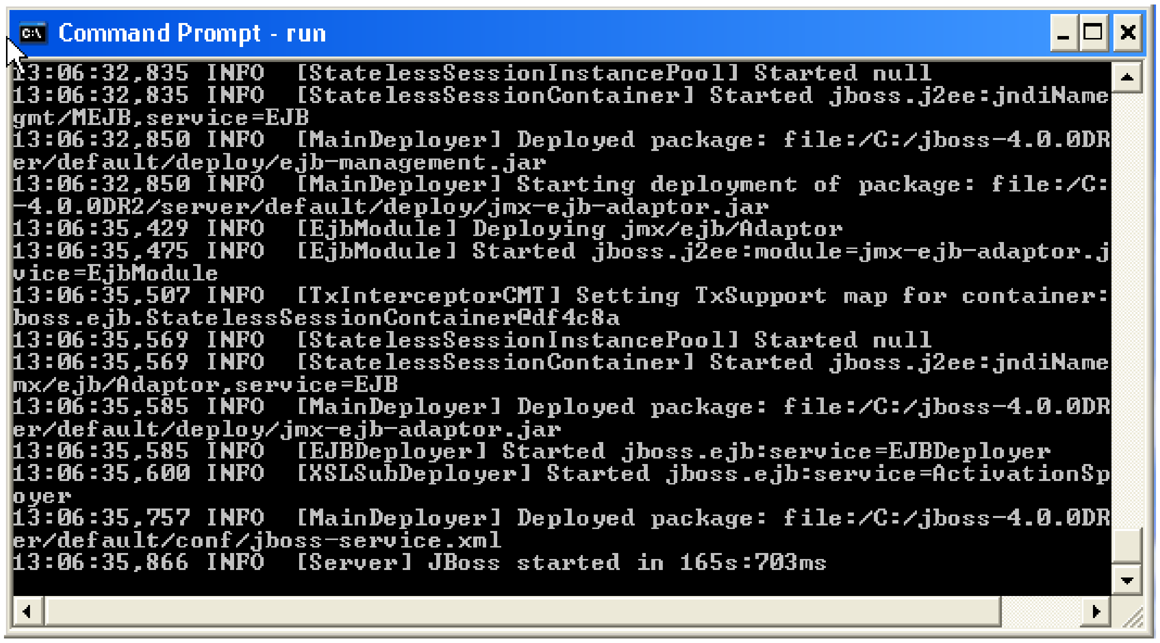
Step 3- build plug-ins.

Open a ***Command Prompt*** window and execute the XTCF-Setup.bat file from the XTCF-Tools folder. Then, change to the C:\XTCF\XTCF-Plugins\xtcf-plugins directory and type **ant**. This will build all of the xtcf plugins, including the evis plugins.

6 WORK FLOW

This workflow describes how the Plugins work on a single computer, but they can be run on separate computers that are pointed to a JMS server.

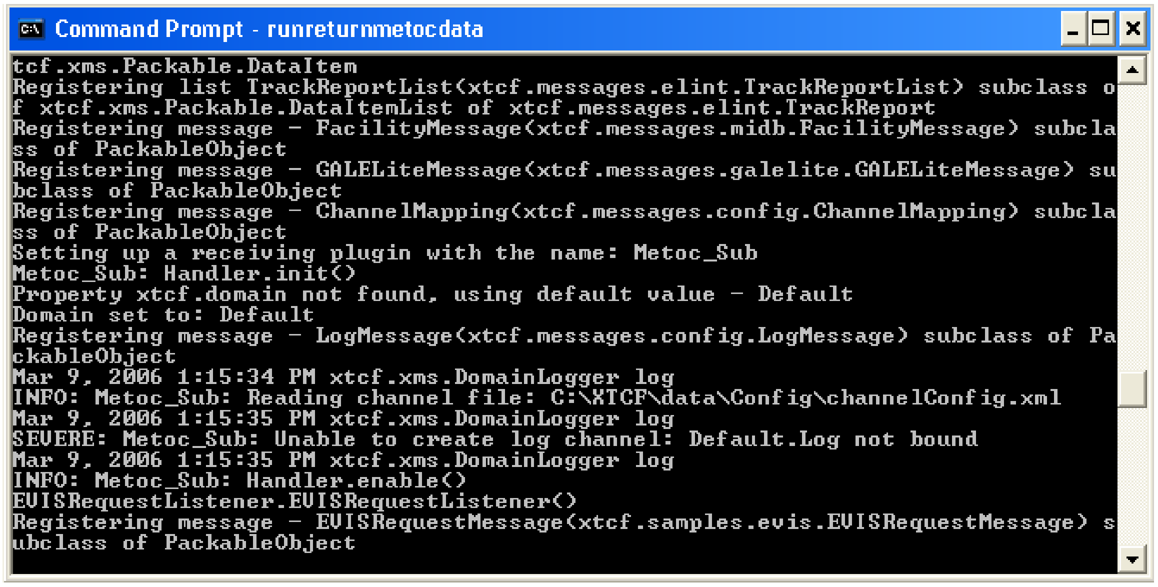
First open three command prompt windows. In the first one start JBoss.



```
13:06:32,835 INFO  [StatelessSessionInstancePool] Started null
13:06:32,835 INFO  [StatelessSessionContainer] Started jboss.j2ee:jndiName
gmt/MEJB,service=EJB
13:06:32,850 INFO  [MainDeployer] Deployed package: file:/C:/jboss-4.0.0DR
er/default/deploy/ejb-management.jar
13:06:32,850 INFO  [MainDeployer] Starting deployment of package: file:/C:
-4.0.0DR2/server/default/deploy/jmx-ejb-adaptor.jar
13:06:35,429 INFO  [EjbModule] Deploying jmx/ejb/Adaptor
13:06:35,475 INFO  [EjbModule] Started jboss.j2ee:module=jmx-ejb-adaptor.j
vice=EjbModule
13:06:35,507 INFO  [TxInterceptorCMT] Setting TxSupport map for container:
boss.ejb.StatelessSessionContainer@df4c8a
13:06:35,569 INFO  [StatelessSessionInstancePool] Started null
13:06:35,569 INFO  [StatelessSessionContainer] Started jboss.j2ee:jndiName
mx/ejb/Adaptor,service=EJB
13:06:35,585 INFO  [MainDeployer] Deployed package: file:/C:/jboss-4.0.0DR
er/default/deploy/jmx-ejb-adaptor.jar
13:06:35,585 INFO  [EJBDeployer] Started jboss.ejb:service=EJBDeployer
13:06:35,600 INFO  [XSLSubDeployer] Started jboss.ejb:service=ActivationSp
oyer
13:06:35,757 INFO  [MainDeployer] Deployed package: file:/C:/jboss-4.0.0DR
er/default/conf/jboss-service.xml
13:06:35,866 INFO  [Server] JBoss started in 165s:703ms
```

Figure 6-1. JBoss Started

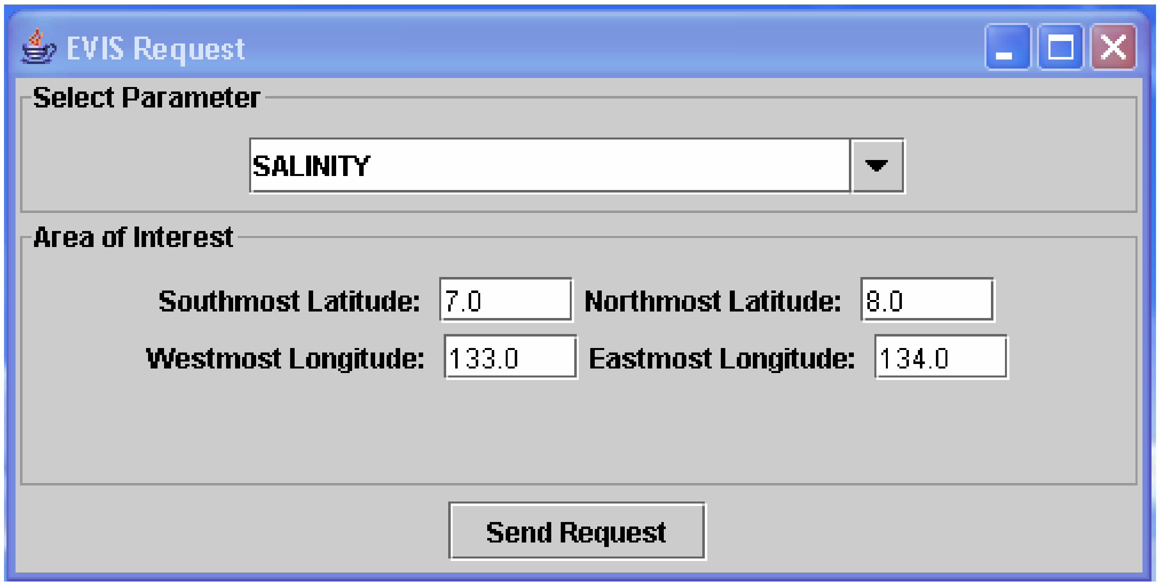
After you see that JBoss has started (Figure 6-1), in one of the other command prompt windows run the batch file that starts the Plug-in that listens for the request for data (Figure 6-2).



```
tcf.xms.Packable.DataItem
Registering list TrackReportList(xtcf.messages.elint.TrackReportList) subclass o
f xtcf.xms.Packable.DataItemList of xtcf.messages.elint.TrackReport
Registering message - FacilityMessage(xtcf.messages.midb.FacilityMessage) subcla
ss of PackableObject
Registering message - GALELiteMessage(xtcf.messages.galelite.GALELiteMessage) su
bclass of PackableObject
Registering message - ChannelMapping(xtcf.messages.config.ChannelMapping) subcla
ss of PackableObject
Setting up a receiving plugin with the name: Metoc_Sub
Metoc_Sub: Handler.init()
Property xtcf.domain not found, using default value - Default
Domain set to: Default
Registering message - LogMessage(xtcf.messages.config.LogMessage) subclass of Pa
ckableObject
Mar 9, 2006 1:15:34 PM xtcf.xms.DomainLogger log
INFO: Metoc_Sub: Reading channel file: C:\XTCF\data\Config\channelConfig.xml
Mar 9, 2006 1:15:35 PM xtcf.xms.DomainLogger log
SEVERE: Metoc_Sub: Unable to create log channel: Default.Log not bound
Mar 9, 2006 1:15:35 PM xtcf.xms.DomainLogger log
INFO: Metoc_Sub: Handler.enable()
EVISRequestListener.EVISRequestListener()
Registering message - EVISRequestMessage(xtcf.samples.evis.EVISRequestMessage) s
ubclass of PackableObject
```

Figure 6-2. Reply Plug-In Listens for Request Messages

Then in the third window run the batch file that starts the Request plug-in and the EVIS Request GUI will pop up (Figure 6-3).



Select Parameter: SALINITY

Southmost Latitude: 7.0   Northmost Latitude: 8.0

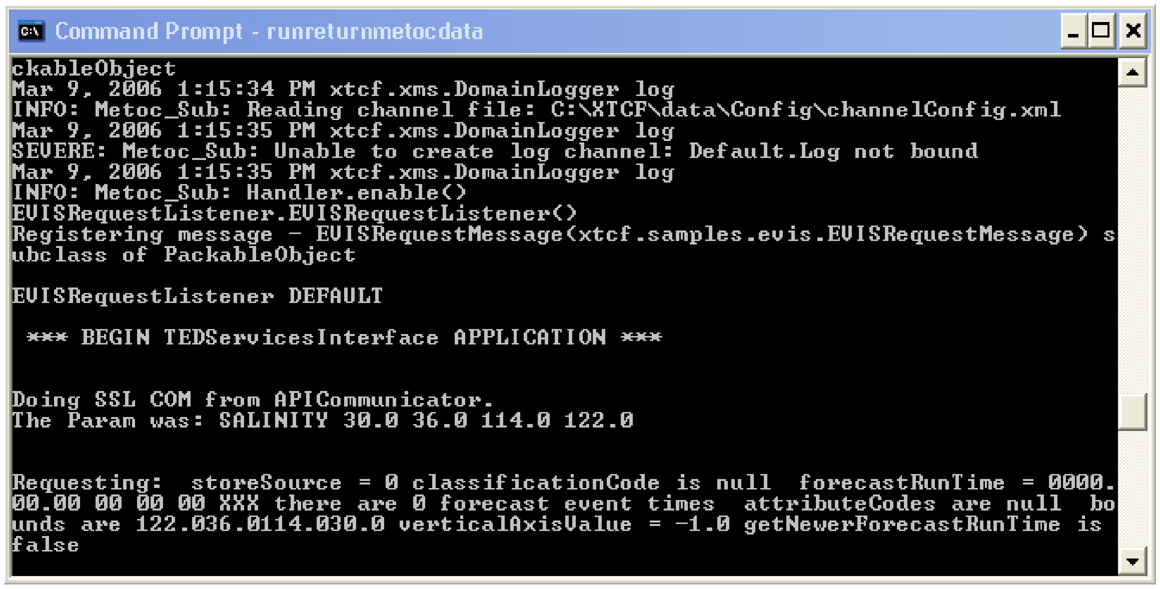
Westmost Longitude: 133.0   Eastmost Longitude: 134.0

Figure 6-3. EVIS Request GUI

The EvisRequestGUI (Figure 6-3) displays a JAVA window that has a dropdown list for the user to select a parameter. The following is a list of the parameters used in this GUI.

ABSOLUTE_VORTICITY_ISOBAR_LEVEL
AIR_TEMPERATURE_HEIGHT_SURFACE
AIR_TEMPERATURE_ISOBAR_LEVEL
AIR_TEMPERATURE_MAX_WIND_LEVEL;
SALINITY
SEA_TEMPERATURE
SIGNIFICANT_WAVE_HEIGHT
SIGNIFICANT_WAVE_PERIOD
SOUND_VELOCITY
TAF
TOTAL_CLOUD_COVER_HIGH_CLOUD_LEVEL
TOTAL_CLOUD_COVER_LOW_CLOUD_LEVEL
TOTAL_CLOUD_COVER_MID_CLOUD_LEVEL
TOTAL_CLOUD_COVER_SKY_COVER
TOTAL_PRECIPITATION_MEAN_SEA_LEVEL
TOTAL_PRECIPITATION_SURFACE

This is not a complete list of parameters that are supported by TEDServices. For a complete list of available parameters, please contact TEDServices and request a copy of the current ParameterNameLookup.txt file. Also, there are 4 input boxes that allow the user to input latitude and longitudes for various areas of interest. All of which are filled with default values when the program is run. Once the user has chosen the correct parameter and AOI, they push the send request button. This calls the EVIS PluginSender to send the request (Figure 6-4).



```
ckableObject
Mar 9, 2006 1:15:34 PM xtcf.xms.DomainLogger log
INFO: Metoc_Sub: Reading channel file: C:\XTCF\data\Config\channelConfig.xml
Mar 9, 2006 1:15:35 PM xtcf.xms.DomainLogger log
SEVERE: Metoc_Sub: Unable to create log channel: Default.Log not bound
Mar 9, 2006 1:15:35 PM xtcf.xms.DomainLogger log
INFO: Metoc_Sub: Handler.enable()
EVISRequestListener.EVISRequestListener()
Registering message - EVISRequestMessage(xtcf.samples.evis.EVISRequestMessage) s
ubclass of PackableObject

EVISRequestListener DEFAULT

 *** BEGIN TEDServicesInterface APPLICATION ***


Doing SSL COM from APICommunicator.
The Param was: SALINITY 30.0 36.0 114.0 122.0


Requesting:  storeSource = 0 classificationCode is null  forecastRunTime = 0000.
00.00 00 00 00 XXX there are 0 forecast event times  attributeCodes are null  bo
unds are 122.036.0114.030.0 verticalAxisValue = -1.0 getNewerForecastRunTime is
false
```

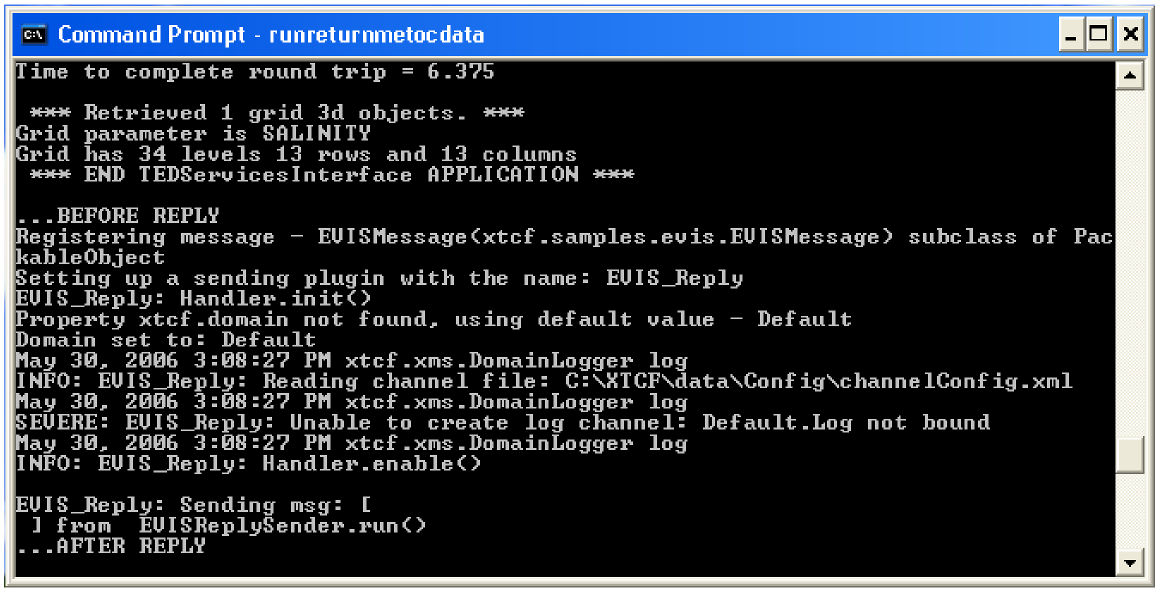
Figure 6-4. Request Plug-In Sends Request Message

The EVISPluginSender sets up the handler and creates a METOCREQUEST topic. It creates an EVISRequestMessage object and copies the parameter and lat/longs into the EVISRequestMessage object. Then it sends the message across the XTCF framework using JBoss.

The Reply Plug-in, which was already started, created a listener and is listening for an EVISRequestMessage on the METOCREQUEST topic. The EVISRequestReceiver sets up a handler and creates a MODASREQUEST channel/topic to receive EVISRequestMessages. Then it creates an EVISRequestListener.

The EvisRequestListener declares a new EVISRequestMessage object. It receives the parameter and lat/longs from the EVISRequestMessage. Then it calls the EVISTestApi method, which sends a request to TEDServices via TEDServices API. This method creates a connection with the TEDServices Gateway and submits the request for grid data for the parameter and AOI specified.

The grid data is returned in a three dimensional array. At the top of figure 6-5 the grid data is returned with salinity values for the requested area of interest. There are 34 levels, which represent various depths of the water and a 13 by 13 matrix for the salinity values of the area at each level.

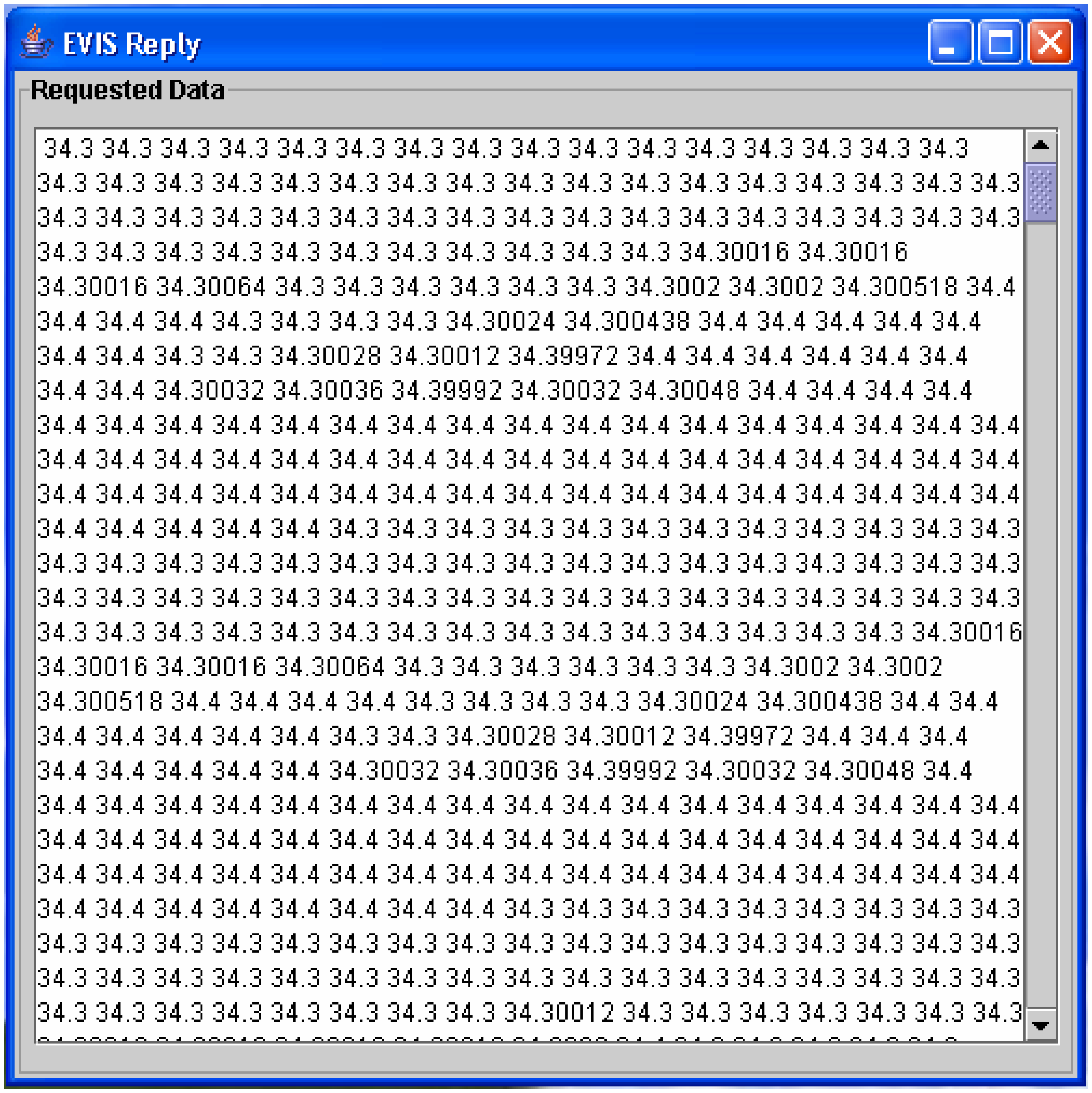


```
Time to complete round trip = 6.375

 *** Retrieved 1 grid 3d objects. ***
Grid parameter is SALINITY
Grid has 34 levels 13 rows and 13 columns
 *** END TEDServicesInterface APPLICATION ***

...BEFORE REPLY
Registering message - EVISMessage(xtcf.samples.evis.EVISMessage) subclass of Pac
kableObject
Setting up a sending plugin with the name: EVIS_Reply
EVIS_Reply: Handler.init()
Property xtcf.domain not found, using default value - Default
Domain set to: Default
May 30, 2006 3:08:27 PM xtcf.xms.DomainLogger log
INFO: EVIS_Reply: Reading channel file: C:\XTCF\data\Config\channelConfig.xml
May 30, 2006 3:08:27 PM xtcf.xms.DomainLogger log
SEVERE: EVIS_Reply: Unable to create log channel: Default.Log not bound
May 30, 2006 3:08:27 PM xtcf.xms.DomainLogger log
INFO: EVIS_Reply: Handler.enable()

EVIS_Reply: Sending msg: [
 ] from  EVISReplySender.run()
...AFTER REPLY
```

Figure 6-5. EVIS Receives and Returns TEDs Data

Once the EvisRequestListener has received the data from TEDServices and prints it to the screen it sends a reply message (EVISMessage object) back to the requestor on the METOCREPLY topic (Figure 6-5).

The computer that made the request for data using the EVIS Request GUI will get a pop window titled EVIS Reply (Figure 6-6) that will display the grid data returned from the request. Also, a file will be created and saved to the requesting computer in the C:\evis\received directory with the returned data.



EVIS Reply

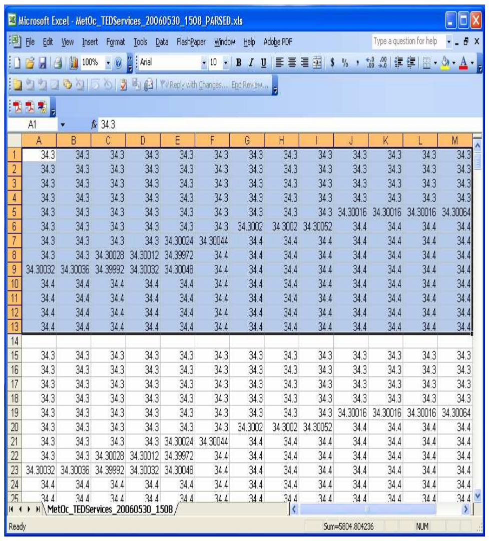
Requested Data

```
34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3
34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3
34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3
34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.30016 34.30016
34.30016 34.30064 34.3 34.3 34.3 34.3 34.3 34.3 34.3002 34.3002 34.300518 34.4
34.4 34.4 34.4 34.3 34.3 34.3 34.3 34.30024 34.300438 34.4 34.4 34.4 34.4 34.4
34.4 34.4 34.3 34.3 34.30028 34.30012 34.39972 34.4 34.4 34.4 34.4 34.4 34.4
34.4 34.4 34.30032 34.30036 34.39992 34.30032 34.30048 34.4 34.4 34.4 34.4
34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4
34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4
34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4
34.4 34.4 34.4 34.4 34.4 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3
34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3
34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3
34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.30016
34.30016 34.30016 34.30064 34.3 34.3 34.3 34.3 34.3 34.3 34.3002 34.3002
34.300518 34.4 34.4 34.4 34.4 34.3 34.3 34.3 34.3 34.30024 34.300438 34.4 34.4
34.4 34.4 34.4 34.4 34.4 34.3 34.3 34.30028 34.30012 34.39972 34.4 34.4 34.4
34.4 34.4 34.4 34.4 34.4 34.30032 34.30036 34.39992 34.30032 34.30048 34.4
34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4
34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4
34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4
34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.4 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3
34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3
34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3
34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.3 34.30012 34.3 34.3 34.3 34.3 34.3 34.3 34.3
```

Figure 6-6. TEDs Data Received and Displayed in GUI

7 DATA

The numerical data that is returned from TEDServices via the plug-ins is in the form of grid data. This is a raw form of data that can be displayed and analyzed by various software tools and programs. For example, the Environmental Visualization (EVIS) capability, which allows users to access weather effect products and enables forecasters the ability to create weather products for mission planning, uses the Grid Analysis and Display System (GrADS). GrADS is an interactive desktop tool, used to manipulate and display grid data. The following example describes how grid data for salinity of a certain AOI can be displayed.

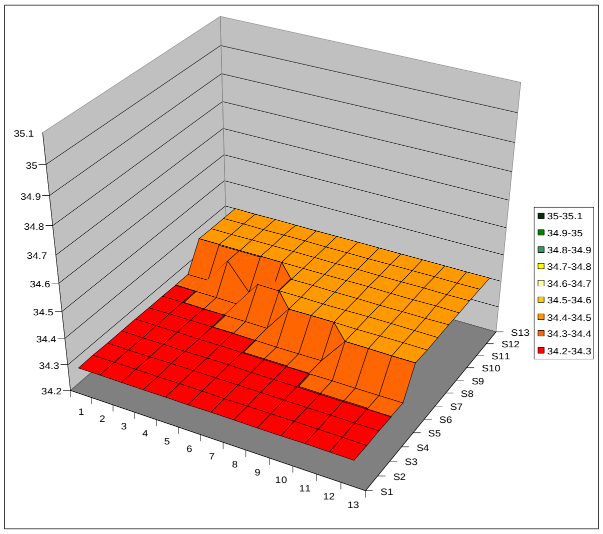

Microsoft Excel - MetOc_TEDServices_20060530_1508_PARSED.xls
File Edit View Insert Format Tools Data FlashPaper Window Help Adobe PDF
Type a question for help
100%
Arial
10
Reply with Changes... End Review...
Ready
NUM

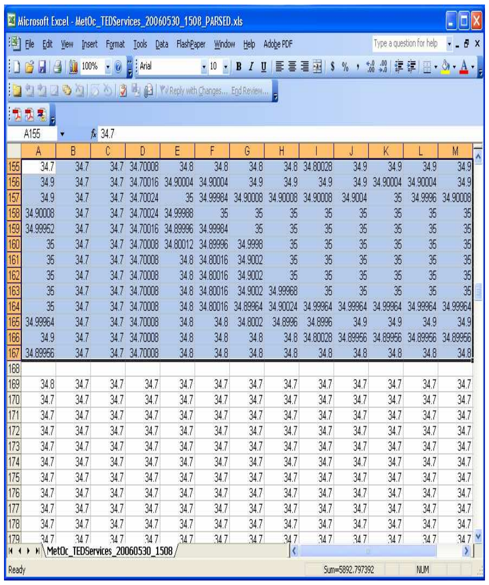
A1 | 34.3

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3
2	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3
3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3
4	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3
5	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.30016	34.30016	34.30016	34.30064
6	34.3	34.3	34.3	34.3	34.3	34.3	34.3002	34.3002	34.30052	34.4	34.4	34.4	34.4
7	34.3	34.3	34.3	34.3	34.30024	34.30044	34.4	34.4	34.4	34.4	34.4	34.4	34.4
8	34.3	34.3	34.30028	34.30012	34.39972	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4
9	34.30032	34.30036	34.39992	34.30032	34.30048	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4
10	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4
11	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4
12	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4
13	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4
14													
15	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3
16	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3
17	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3
18	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3
19	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.3	34.30016	34.30016	34.30016	34.30064
20	34.3	34.3	34.3	34.3	34.3	34.3	34.3002	34.3002	34.30052	34.4	34.4	34.4	34.4
21	34.3	34.3	34.3	34.3	34.30024	34.30044	34.4	34.4	34.4	34.4	34.4	34.4	34.4
22	34.3	34.3	34.30028	34.30012	34.39972	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4
23	34.30032	34.30036	34.39992	34.30032	34.30048	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4
24	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4
25	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4	34.4

MetOc_TEDServices_20060530_1508

Sum=5804.804236




Microsoft Excel - MetOc_TEDServices_20060530_1508_PARSED.xls
File Edit View Insert Format Tools Data FlashPaper Window Help Adobe PDF
Type a question for help
100%
Arial
10
Reply with Changes... End Review...
Ready
NUM

A155 | 34.7

	A	B	C	D	E	F	G	H	I	J	K	L	M
155	34.7	34.7	34.7	34.70008	34.8	34.8	34.8	34.8	34.80028	34.9	34.9	34.9	34.9
156	34.9	34.7	34.7	34.70016	34.90004	34.90004	34.9	34.9	34.9	34.9	34.90004	34.90004	34.9
157	34.9	34.7	34.7	34.70024	35	34.99984	34.90008	34.90008	34.90008	34.9004	35	34.9996	34.90008
158	34.90008	34.7	34.7	34.70024	34.99988	35	35	35	35	35	35	35	35
159	34.99952	34.7	34.7	34.70016	34.89996	34.99984	35	35	35	35	35	35	35
160	35	34.7	34.7	34.70008	34.80012	34.89996	34.9998	35	35	35	35	35	35
161	35	34.7	34.7	34.70008	34.8	34.80016	34.9002	35	35	35	35	35	35
162	35	34.7	34.7	34.70008	34.8	34.80016	34.9002	35	35	35	35	35	35
163	35	34.7	34.7	34.70008	34.8	34.80016	34.9002	34.99968	35	35	35	35	35
164	35	34.7	34.7	34.70008	34.8	34.80016	34.89964	34.90024	34.99964	34.99964	34.99964	34.99964	34.99964
165	34.99964	34.7	34.7	34.70008	34.8	34.8	34.8002	34.8996	34.8996	34.9	34.9	34.9	34.9
166	34.9	34.7	34.7	34.70008	34.8	34.8	34.8	34.8	34.80028	34.89956	34.89956	34.89956	34.89956
167	34.89956	34.7	34.7	34.70008	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8	34.8
168													
169	34.8	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7
170	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7
171	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7
172	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7
173	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7
174	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7
175	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7
176	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7
177	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7
178	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7
179	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7	34.7

MetOc_TEDServices_20060530_1508

Sum=5892.797392

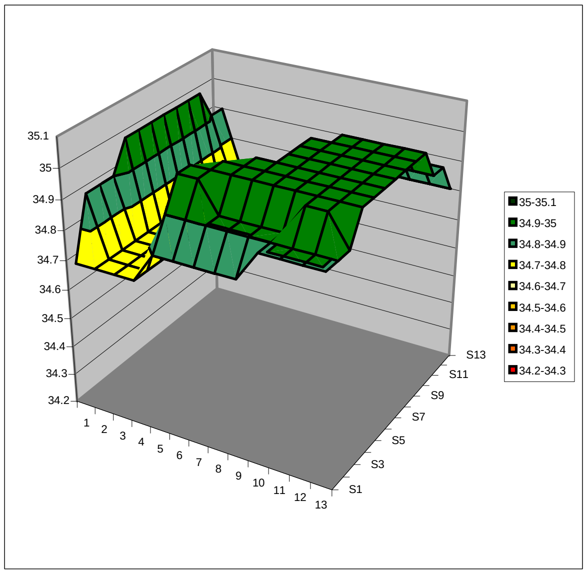


Figure 7-1. Grid Data Display

The salinity of the ocean is a good example of how grid data is used. Salinity is measured as the conductivity ratio of sea water to a standard solution. The values of salinity constantly change as you move from one area to another forming a continuous spatial gradient. The left-side of figure 7-1 shows the grid data and highlighted sub-sets of salinity values used to depict the surfaces shown on the right-side.

Grid data is stored as an organized set of values in a matrix that is geo-registered. In this example, each grid cell identifies a specific location and contains a map value representing its salinity ratio. Also, grid data can represent various levels of the atmosphere. In this case, the levels refer to the depth of the water in meters. The data in the top of figure 7-1 is close to the surface of the water and the data at the bottom is about 12 meters below the surface. If all of the data from this example were shown you would see 34 (13x13) matrices representing the salinity values for an area of interest at various depths in meters.

8 TROUBLESHOOTING TIPS

The following are some issues to keep in mind when you are trouble shooting your code.

- Remember that when you compile the XTCF source code, XTCF_SRC, it overwrites xtcf\bin. Also, it deletes and recreates xtcf\lib
- Ensure that the TEDServices jar file was added to your class path (i.e. xtcf\lib\tedservices\).
- Check that Tedservices.cert is in the java keystore.
- Double check that Jboss is configured correctly and ensure that the Topics or Queues are set up correctly.
- Check to ensure that all environment variables are set as follows:
  - `XTCF_HOME -` Location of XTCF (defaults to `XTCF`)
  - `JMS_HOME -` Location of JMS (defaults to `XTCF-Tools`)
  - `JAVA_HOME -` Location of Java SDK (defaults to `XTCF-Tools`)
  - `ANT_HOME -` Location of Ant (defaults to `XTCF-Tools`)
  - `JBOSS_HOME -` Location of JBoss when this is the JMS provider

9 REFERENCES

[1] Berry, J.K. (2006). Map Analysis: Procedures and Applications in GIS Modeling. BASIS Press. Retrieved August 21, 2006, from http://www.innovativegis.com/basis/MapAnalysis/Topic18/Topic18.htm

[2] Chapell, D.A. & Monson-Haefel, R. (2001). Java Message Service. Sebastopol, CA: O'Reilly Media.

[3] Filanowitcz, B. Knowledge Superiority and Assurance Future Naval Capability Dynamic Command and Control Continuum Sea Trial (Technical Document 3193). SPAWAR Systems Center San Diego.

[4] Harold, E.R. & Means, W.S. (2004). XML in a Nutshell: A Desktop Reference (3rd ed.). Sebastopol, CA: O'Reilly Media.

[5] Naval Research Laboratory. TEDServices Client Application Developer Tool Kit [Computer Software and Manual]. Retrieved August 27, 2004, from https://teds.navy.mil/

[6] Net-Centric Enterprise Solutions for Interoperability. (2005) NESI Part 5: Net-Centric Developers Guide. Retrieved May 22, 2006, from http://nesipublic.spawar.navy.mil/

[7] Space and Naval Warfare Systems Center. Developer SOFTWARE INSTALLATION GUIDE FOR THE EXTENSIBLE TACTICAL C4I FRAMEWORK (XTCF)Version 3.0 27 February 2004

[8] Space and Naval Warfare Systems Center. DEVELOPER GUIDE FOR THE EXTENSIBLE TACTICAL C4I FRAMEWORK (XTCF) SOFTWARE DEVELOPMENT KIT (SDK) Version 3.0 27 February 2004

APPENDIX A. ACRONYM LIST

ABA	Advanced Battlespace Awareness
ACINT	Acoustic Intelligence
ACTD	Advanced Concept Technology Demonstration
ADSI	Air Defense Systems Integration
AOI	Area of Interest
API	Application Programmer Interface
C2	Command and Control
C4I	Command, Control, Communications, Computers, and Intelligence
CD	Compact Disk
COE	Common Operating Environment
COI	Community of Interest
DID	Data Item Description
DISA	Defense Information Systems Agency
EVIS	Environmental Visualization
FMACL	Fleet METOC Advanced Concepts Laboratory
GCCS	Global Command and Control System
GCCS-A	Global Command and Control System-Army
GCCS-M	Global Command and Control System-Maritime
GrADS	Grid Analysis and Display System
GUI	Graphical User Interface
HTTP	Hyper Text Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers, Inc.
J2EE	Java 2 Platform Enterprise Edition
JBDC	Java Database Connectivity
JMS	Java Message Service
JNDI	Java Naming and Directory Service
LAN	Local Area Network
MetOc	Metrological and Oceanographic
ONR	Office of Naval Research
RPC	Remote Procedure Call
SDK	Software Development Kit
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SPAWAR	Space and Naval Warfare Systems Command
SSC	SPAWAR Systems Center
SSL	Secure Socket Layer
XML	eXtensible Markup Language
XSD	XML Schema Document
XTCF	Extensible Tactical C4I Framework

APPENDIX B. EVISMESSAGE.JAVA

```
//  EVISMessage.java
//  This file was automatically generated by XTCF codegen Version 2.0
//
package xtcf.samples.evis;
import javax.xml.namespace.QName;
import xtcf.xms.Packable.PackableObject;
import xtcf.xms.Packable.PackableSerializationException;
import xtcf.xms.Packable.MessageData;
import xtcf.xms.Packable.MessageItem;

/**
*  The EVISMessage class objects represent Java runtime
*  instantiations of EVISMessage XTCF messages.
*
*/

public class EVISMessage extends PackableObject
{
  public EVISMessage()
  {
  }

  public EVISMessage( String fileNameValue, byte[] fileContentValue )
  {
    this();
    setFileName( fileNameValue );
    setFileContent( fileContentValue );
  }

  // static message item list
  private static MessageItem[] messageItems = {
    new MessageItem( EVISMessage.class, "FileName",
DbDataType.DBT_STRING, 0, "", null, false ),
    new MessageItem( EVISMessage.class, "FileContent",
DbDataType.DBT_BYTE_ARRAY,
PackableObject.MessageItemFlags.MIF_HEX_ENCODE, "", null, false ),

  };


  static String comments = "";


  private static MessageData messageData = new MessageData(
EVISMessage.class, messageItems, 0, "EVISMessage", "PackableObject",
comments, "urn:evis.samples.xtcf", "EVISMessage.xsd" , false );


  static {

    int n = 0;

    messageItems[n].setMinOccurs( 1 );

    messageItems[n].setMaxOccurs( 1 );

    messageItems[n].setItemQName( new QName( "", "FileName" ) );

    messageItems[n].setTypeQName( new QName(
"http://www.w3.org/2001/XMLSchema", "string" ) );

    n++;


    messageItems[n].setMinOccurs( 1 );

    messageItems[n].setMaxOccurs( 1 );

    messageItems[n].setItemQName( new QName( "", "FileContent" ) );

    messageItems[n].setTypeQName( new QName(
"http://www.w3.org/2001/XMLSchema", "hexBinary" ) );

    n++;

  };


  /**

   *  Get full description of message, including array of items.

   */

  public MessageData getMessageData()

  {

    return messageData;

  }


  /**

   *  Get array of message items.

   */
```

```
public MessageItem[] getMessageItems()
{
  return messageItems;
}

public void setFileName( String value )
{
  try {
    this.setValue( "FileName", value );
  }
  catch ( PackableSerializationException pse )
  {
    System.out.println( pse.getMessage() );
    pse.printStackTrace();
  }
}

public String getFileName()
{
  try {
    return ( String )this.getValue( "FileName" );
  }
  catch ( PackableSerializationException pse )
  {
    System.out.println( pse.getMessage() );
    pse.printStackTrace();
  }
  return "";
}

public void setFileContent( byte[] value )
{
  try {
    this.setValue( "FileContent", value );
  }
  catch ( PackableSerializationException pse )
  {
        System.out.println( pse.getMessage() );
        pse.printStackTrace();
      }
    }

    public byte[] getFileContent()
    {
      try {
        return ( byte[] )this.getValue( "FileContent" );
      }
      catch ( PackableSerializationException pse )
      {
        System.out.println( pse.getMessage() );
        pse.printStackTrace();
      }
      return null;
    }
  }
```